Journal of Zhejiang University SCIENCE
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

A text to speech interface for Universal Digital Library

PRAHALLAD Kishore[1,2], BLACK Alan[1]

([1]*Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15217, USA*)

([2]*International Institute of Information Technology, Hyderabad, AP, 500019, India*)

E-mail: skishore@cs.cmu.edu; awb@cs.cmu.edu

Received Aug. 5, 2005; revision accepted Sept. 10, 2005

Abstract: The objective of Universal Digital Library (UDL) is to capture all books in digital format. A text to speech (TTS) interface for UDL portal would enable access to the digital content in voice mode, and also provide access to the digital content for illiterate and vision-impaired people. Our work focuses on design and implementation of text to speech interface for UDL portal primarily for Indian languages. This paper is aimed at identifying the issues involved in integrating text to speech system into UDL portal and describes the development process of Hindi, Telugu and Tamil voices under Festvox framework using unit selection techniques. We demonstrate the quality of the Tamil and Telugu voices and lay out the plan for integrating the TTS into the UDL portal.

Key words: Text to speech (TTS), Indian language, Universal Digital Library (UDL)
doi:10.1631/jzus.2005.A1229 **Document code:** A **CLC number:** TP391

INTRODUCTION

The objective of Universal Digital Library (UDL) is to capture all books in digital format (Universal Digital Library, 2005; Digital Library of India, 2005). Most of the digital information present in the digital world is accessible to a few who can read or understand a particular language. Language technologies can provide solutions in the form of natural interfaces so that digital content can reach the masses and facilitate the exchange of information across different people speaking different languages.

These technologies play a crucial role in multi-lingual societies such as India which has about 1652 dialects/native languages. While Hindi written in Devanagari script, is the official language, the other 17 languages recognized by the constitution of India are: (1) Assamese; (2) Tamil; (3) Malayalam; (4) Gujarati; (5) Telugu; (6) Oriya; (7) Urdu; (8) Bengali; (9) Sanskrit; (10) Kashmiri; (11) Sindhi; (12) Punjabi; (13) Konkani; (14) Marathi; (15) Manipuri; (16) Kannada; and (17) Nepali. Seamless integration of speech recognition, machine translation and speech synthesis systems could facilitate the exchange of information between two people speaking two different languages. Our overall goal is to develop speech recognition and speech synthesis systems for most of the above mentioned languages (Prahallad and Black, 2003; Prahallad *et al*., 2003).

For UDL, a voice interface would enable access to the digital content in voice mode, and thus provide access to the digital content to illiterate and vision-impaired people. In the context of Indian scenario, while illiterate people cannot read/write, most of them are bi-lingual speakers. A voice activated portal would help these bilingual speakers to understand the information present in one of the known languages. The focus of this work is to design and implement a text to speech (TTS) interface for UDL portal primarily for Indian languages. The scope of this paper is to address the issues involved in building such an interface and demonstrate the text to speech systems for Hindi, Telugu and Tamil languages.

This paper is organized as follows: Section 2 explains the data-flow in text to speech enabled UDL portal. Section 3 discusses the issues involved in integration of TTS into the UDL portal. Sections 4 and 5 describe the text normalization and speech

generation components of a TTS system. Sections 6 and 7 describe the development process of Indian language voices under the framework of Festvox.

DATA FLOW IN TEXT TO SPEECH ENABLED UDL PORTAL

Fig.1 depicts the data flow in an UDL portal which includes a text to speech system.

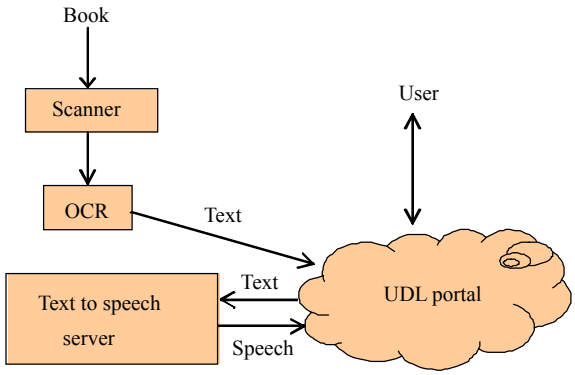

Fig.1 Data-flow in an UDL portal integrated with text to speech system

The process of digitization begins with the book being scanned page by page. Thus a digitized book is a series of images where each image corresponds to a page in the book. Each digitized page is processed using Optical Character Recognition (OCR) to obtain text in ASCII or Unicode format. The digitized text is stored in the UDL portal. Upon a request from the user, this text is sent to a text to speech (TTS) system for conversion into a speech signal.

ISSUES IN INTEGRATING TTS INTO THE UDL PORTAL

While the conceptual diagram described in Section 2 explains the data-flow, the implementation aspects of it need to address the following issues.

User interface

Given the text to speech capability to the UDL portal a user may want to:

(1) Listen to the entire book by clicking on a icon;

(2) Listen to a set of pages by giving the page numbers;

(3) Listen to a page by clicking on a icon;

(4) Listen to a paragraph by selecting the text.

If a book has to be read, the TTS system should be able to handle the title, author, publisher (it may need to construct meaningful sentences to say who is the author, publisher, etc.), preface, table of contents, table of figures, acknowledgements, etc. Whether the user wants to know about the author, publisher details or just the content has to be taken into account. If a paragraph has to be read, the portal should be able to support the select and speak option.

Specification of TTS systems

1. Noisy input

Text to speech systems typically expect an error-free text as input. However, the accuracy of an OCR, even in some of the most developed languages, is hindered by the bad quality of the images. This is particularly true for older books and those that use ancient fonts for which the OCR is not tuned. Even the best OCR accuracy of the order of 98% may not be acceptable in some cases. A spoken form of a noisy text may have wrong pronunciations for some of the words or sometimes may lead to a different interpretation of a word or a sentence.

2. Formatted text

The output of OCR may be in generic XML tags or HTML tags or local-defined tags to define formatting of the text. The TTS system should be able to handle these tags and extract the relevant text for synthesis. Moreover, the text may be encoded in Unicode (UTF-8) format or in a transliteration scheme used to key-in the non-English languages.

3. Language identification

As we are dealing with Universal library, the TTS system should support multiple languages. Given the framework of Festvox it is easy to provide multi-lingual support for the UDL. However to switch to a particular language, it is essential to know the language either from the input text or from the arguments passed to the TTS from the portal. It is easy to identify the language from the input text if it is encoded in Unicode. This avoids the extra effort from the portal to get the language information from the meta-data of the book.

Mode of delivery

Another issue is the mode of delivery and the response time of the TTS system. Given a book to be read, the TTS system would synthesize a first few sentences and stream it back to the user in real-time. While the user is listening to this text, the system runs a parallel thread to synthesize part of the text. However, if the user wants to store the entire audio file (for a pod-cast), then the response of the TTS system would be longer, sometimes may be in hours depending on the size of the book.

TEXT NORMALIZATION

Given the integration of TTS into the UDL portal, the following sections describe the steps involved in generating speech from text. There are two major components in a text to speech system: (1) text normalization and (2) speech generation. Before we discuss the issues related to text processing, let us briefly discuss the nature of the Indian languages scripts for which the synthesis systems are built.

The basic units of the writing system in Indian languages are Aksharas, which are orthographic representations of speech sounds. An Akshara in Indian language scripts is a syllable and can be typically of the following form: V, CV, CCV and CCCV where C is a consonant and V is a vowel. All Indian language scripts have a common phonetic base, and a universal phone set consisting of about 35 consonants and about 18 vowels. The pronunciation of these scripts is almost straightforward. There is more or less one to one correspondence between what is written and what is spoken. However, in languages such as Hindi and Bengali the inherent vowel (short /a/) associated with a consonant is not pronounced depending on the context. It is referred to as inherent vowel suppression or schwa deletion.

Unicode to IT3 conversion

As discussed in Section 3, the input text given to a TTS system is better off being encoded in UTF-8 format ((The Unicode Consortium, 2003)). However, for the sake of human-readable format and for debugging purposes, the synthesis engines for non-English languages are built using a transliteration scheme. For our case, we use IT3 transliteration scheme. This scheme is phonetic in nature and a one to one mapping could be built from Unicode to IT3 code. The input text is processed through Unicode to IT3 mapping module to produce text in IT3 format.

Mapping of non-standard words to standard words

In practice, an input text from books, news articles consists of standard words (whose entries could be found in the pronunciation dictionary) and non-standard words such as initials, digits, symbols and abbreviations. Mapping of non-standard words to a set of standard words depends on the context, and is a non-trivial problem. For example, digit 120 has to be expanded to /nuut'aa iravai/, "Rs 3005412" to /muppia laqs-alaa aidu velaa, nalagu van'dalaa pannen'd'u ruupayalu/, and "Tel: 3005412" to /phon nambaru, muud'u sunnaa sunnaa, aidu nalagu okati rend'u/. Similarly punctuation characters and their combinations such as :, >, !, -, $, #, %, / which may be encountered in the cases of ratios, percentages, comparisons have to be mapped to a set of standard words according to the context.

Other such situations include initials, company names, street address, initials, titles, non-native words such as bank, computer etc.

Standard words to phone sequence

Generation of sequence of phone units for a given standard word is referred to as letter to sound rules. The complexity of these rules and their derivation depends on the nature of the language. For languages such as English, a pronunciation dictionary of about 125 000 words is used along with a set of letter to sound rules to handle unseen words. For Indian languages such as Telugu the letter to sound rules are relatively easy due to their phonetic nature, i.e., there is a fairly good correspondence between what is written and what is spoken.

However, for some of the Indian languages such as Hindi and Bengali, the rules for mapping of letter to sound are not so straightforward. In Hindi, the inherent schwa associated with a consonant is suppressed depending on the context. For example, words such as /kamala/ and /dilachaspa/ are pronounced as /kamal/ and /dilchasp/ respectively. To build a good text processing module and thus to generate natural sounding speech synthesis in Hindi

and other such languages, understanding of this phenomenon is important. We are now using a small set of heuristic rules and working on machine learning techniques for schwa deletion.

SPEECH GENERATION

Given the sequence of phones (or sound units), the objective of the speech generation component is to synthesize the acoustic waveform. Speech generation has been attempted by articulatory model based techniques and parametric based techniques. While the articulatory models suffer from lack of adequate modelling of articulator motion, the parametric models require a large number of rules to manifest coarticulation and prosody. An alternative solution is to concatenate the recorded speech segments. The inventory of these recorded speech segments is limited to a small set of units which have sufficient coarticulation such as diphones and a set of rules were used to manipulate the prosody.

Current state-of-the-art speech synthesis systems generate natural sounding speech by using an inventory of large number of speech units with differing prosody. Storage of large number of units and their retrieval in real time is feasible due to availability of cheap memory and computation power. The approach of using an inventory of speech units is referred to as unit selection approach and can also be referred to as data-driven approach or example-based approach for speech synthesis (Black and Taylor, 1997). The issues related to the unit selection speech synthesis system are: (1) choice of unit size (Prahallad and Black, 2003; Prahallad *et al*., 2003), (2) generation of speech database and (3) criteria for selection of a unit.

The objective criteria for selection of a unit depends on how well it matches with the input specification and how well it matches with the other units in the sequence. Costs are associated for mismatch with the input specification and with other units in sequence, and are referred to as target cost and concatenation cost respectively. A unit which minimizes the cost of target and concatenation cost is selected from the speech database.

BUILDING INDIAN LANGUAGE VOICES

To build a voice in a new language, the steps involved are as follows (Black and Lenzo, 2000):

(1) Defining the phone set of the language;

(2) Incorporation of letter-to-sound rules;

(3) Incorporation of syllabification rules;

(4) Assignment of stress patterns to the syllables in the word;

(5) Selection of text to be recorded;

(6) Recording of speech database;

(7) Labelling the speech database;

(8) Extraction of pitch markers and Mel-frequency cepstral coefficients;

(9) Building the units' database by clustering algorithm;

(10) Fine tuning the parameters such as pitch markers, clustering the units by tagging them with more phonemic context, etc.

In defining the phone set for Indian languages, we use IT3 transliteration scheme to transliterate the Indian language scripts onto the machine.

Letter to sound rules, syllabification and stress patterns

Letter-to-sound rules are almost straightforward in Indian languages, as they are phonetic in nature. We almost speak what we write, and hence generally the necessity of a pronunciation dictionary does not arise in our case. At present, a set of manually written rules are used to obtain the pronunciation. The pronunciation for a Telugu word such as nagaran' (town) in terms of phones marked with syllable boundaries can be written as ((n a) 1) ((g a) 0) ((r a n') 0). As the characters in Indian language are close to a syllable, clustering C*VC* can be done easily using simple rules. Here C is a consonant and V is a vowel. Syllable boundaries are marked at the vowel positions.

For stress assignment, the primary stress is associated with the first syllable and secondary stress with the remaining syllables in the word. The integer "1" assigned to first syllable in the word nagaran' indicates the primary stress associated with it. Letter to sound rules, syllabification rules and assignment of stress patterns for a new language can be implemented easily in Festvox. The architecture of Festival synthesis engine allows these rules to be written in Scheme, so that they get loaded at the runtime, essentially avoiding recompilation of the core code for every new language (Black *et al*., 1998; Black and Lenzo, 2000).

Generation of unit selection database

The generation of unit selection database consists of selection of text to be recorded (Step 5) and the recording of the selected sentences by a voice talent (Step 6).

1. Selection of text

The quality of unit selection voice is inherently bound to speech database from which the units are selected. It is important to have an optimal speech corpus balanced in terms of phonetic coverage and the diversity in the realizations of the units. In this work, speech databases are generated from a set of sentences selected from a large text corpus available in Indian languages. The text selection algorithm here primarily ensures the coverage of high frequency trigrams. The objective of choosing the trigrams as the unit for coverage is to enable the developed speech databases to support non-uniform unit based synthesis algorithm which essentially uses trigrams, bigrams and unigrams, etc. (development of such algorithms is not discussed here). The actual algorithm used to select the sentences is shown below:

(1) Given a large data collection, use font/converters and output the text in IT3 transliteration.

(2) Perform bigram, trigram and 4-gram analysis on the text corpus using CMU Language Model toolkit. Analysis of trigram and 4-gram files will reveal whether the text corpus consists of any high frequency anomalies such as misspelled words, etc. Sometimes due to errors in the Web pages, the blank spaces are found at the syllables as apposed to words leading to high frequency syllables as the top entries. These anomalies can be sorted out by cleaning (either manually or by program) the text corpus. The common mistakes and the corresponding correct answers are also noted so that they can be used as a correcting-table during synthesis.

(3) The corrections done to the misspelled words are incorporated in the text corpus and trigram analysis is performed again.

(4) Select the top n (2000/3000) high frequency trigrams and use them to select a set of sentences from the text corpus. A sentence in the corpus is selected, if it covers at least one new trigram.

2. Recording of the speech database

The recording environment and the voice talent (speaker who speaks the selected sentences) also affect the quality of the synthesized speech. To have better quality speech, the recordings are done either in professional recording studio or in recording sound booth available in academic institutions.

Labeling and building clusters

Once the speech data is recorded, it has to be labelled at the phone (segment) level. To perform this labelling we use full acoustic models built using Sphinx recognition system. This process builds tri-phone acoustic models from the recorded speech data and the corresponding transcription. Once the models are trained the Viterbi algorithm is used to find the alignment and thus the corresponding time stamps are obtained.

Festvox is used to extract pitch markers and Mel-cepstral coefficients, and then to build a decision tree for each unit (phone) based on questions concerning the phonemic and prosodic context of that unit. Once the clusters are built, the final stage dumps all the units into a catalogue file thus completing the process of voice building.

TELUGU AND TAMIL VOICES

Following the approach discussed in Section 6, we built Telugu and Tamil voices. To build the Telugu voice, we extracted a set of 1637 sentences as explained in Section 6.2.1. To obtain these sentences, we used 5 million words of newspaper text corpus extracted from the worldwide Web. The selected sentences were 10~15 words long and covered the top 1807 high frequency trigrams and 1843 high frequency syllables. A female speaker recorded these 1637 sentences in the sound recording booth at Carnegie Mellon University, USA. These sentences were recorded in five sessions spanned over three days.

We followed a similar approach to build a Tamil voice. To build this voice, we collected Tamil text corpus (0.3 million sentences) from a news portal. This corpus has 2.7 million words and 4302 syllables. As discussed in Section 6.2.1, we selected 2394 sentences covering 25769 words and 2392 high frequency syllables. The selected sentences were recorded by a female native Tamil speaker, in a recording studio, uttered the sentences into a stand mounted microphone placed in front of her. The

speech data were recorded at 44 kHz, mono channel at 16 bits per sample. After the recording it was down sampled to 16 kHz for further processing. This recording process yields 8 h of speech.

The recorded speech in Telugu and Tamil was labelled using full acoustic models built for these voices as explained in Section 6.3, and full fledged voices were created for Telugu and Tamil.

CONCLUSION AND FUTURE WORK

The aim of this work was to identify the issues in the integration of TTS into the UDL portal. A successful integration of a TTS system into the UDL needs to address the following aspects:

(1) User interface: What a user is likely to hear: page(s)/book/paragraph;

(2) Mode of delivery: Support for streaming as well as for downloading the entire speech file;

(3) Real-time response of the TTS system;

(4) Markup text: The output of OCR should produce the output text using a standard markup language which can be handled/interpreted by the TTS system;

(5) Use of Unicode encoding scheme for storing the OCR-ed text;

(6) Language identification from the text.

For a successful integration of TTS into the UDL portal, there should be synergy between the user interface and the capabilities of a text to speech system, standards used to store the output of OCR and the input specifications of a text to speech system. In this paper, we have also explained the process of building voices in Telugu and Tamil.

Our future work involves: (1) Development of Hindi voice; (2) Getting better understanding of the user interface aspects of TTS for UDL; (3) Identification of the standard tags to be used to store the OCR output and (4) Software development for integration of TTS into the UDL portal.

References

Black, A., Taylor, P., 1997. Automatically Clustering Similar Units for Unit Selection in Speech Synthesis. Eurospeech97, Rhodes, Greece, **2**:601-604.

Black, A., Lenzo, K., 2000. Building Voices in the Festival Speech Synthesis System. http://festvox.org/bsv/.

Black, A., Taylor, P., Caley, R., 1998. The Festival Speech Synthesis System. http://festvox.org/festival.

Digital Library of India, 2005. http://dli.iiit.net.

Prahallad, K., Black, A., 2003. Unit Size in Unit Selection Speech Synthesis. Proceedings from Eurospeech, Geneva, Switzerland.

Prahallad, K., Black, A., Kumar, R., Sangal, R., 2003. Experiments with Unit Selection Speech Databases for Indian Languages. National Seminar on Language Technology Tools: Implementation of Telugu. Hyderabad, India.

The Unicode Consortium, 2003. http://www.unicode.org.

Universal Digital Library, 2005. http://tera-3.ul.cs.cmu.edu/hci/.